PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2004-004341

(43)Date of publication of application : 08.01.2004

(51)Int.Cl. G09C 1/00

(21)Application number : 2002-160231

(22)Date of filing : 31.05.2002

(71) Applicant : TOSHIBA CORP

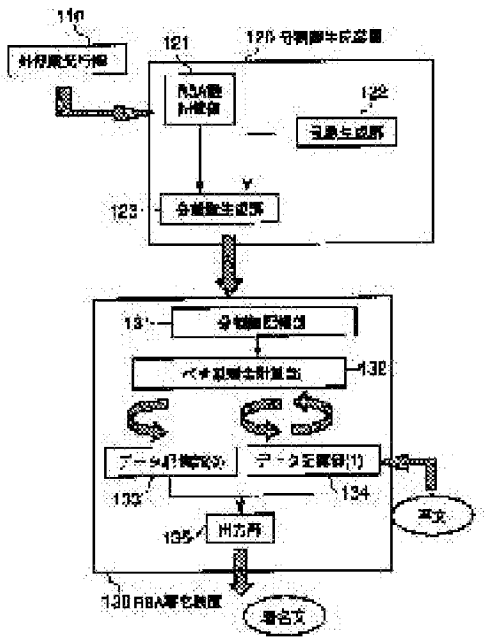
(72)Inventor : TOMOE HIROKI
KAWAMURA SHINICHI
SHINPO ATSUSHI

(54) APPARATUS AND METHOD FOR MODULO EXPONENTIATION CALCULATION, AND PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a modulo exponentiation calculation apparatus which realizes prevention of secret key leakage due to power analysis attack to a public key ciphering system.

SOLUTION: A divided key generation part 120 generates k-number of divided secret keys from the secret key (d) of the public key ciphering system and generates x-number of dummy keys by a random number and a key index consisting of k-number of 1 and x-number of 0. An RSA signature device 130 writes a plaintext M as an initial value in data storage part (0), (1), performs modulo exponentiation calculation by using corresponding single divided secret key with the value of the data storage part (1) as an input when the key index is 1, stores the result in the data storage part (1), performs modulo exponentiation calculation by using corresponding single dummy key with the value of the data storage part (0) as an input when the key index is 0, and stores the result in the data storage part (0). The value of the data storage part (1) after processing is a signature sentence. Even if the divided secret keys are leaked due to power analysis attack, it is difficult to obtain the secret key (d).



LEGAL STATUS

[Date of request for examination]	17.09.2003
[Date of sending the examiner's decision of rejection]	03.10.2006
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	
[Date of registration]	
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2004-4341
(P2004-4341A)
(43)公開日　平成16年1月8日(2004.1.8)

(51)Int.Cl.[7]	F I			テーマコード（参考）
G09C 1/00	G09C	1/00	650A	5J104
	G09C	1/00	620A	

審査請求 有　請求項の数 23　OL　（全 29 頁）

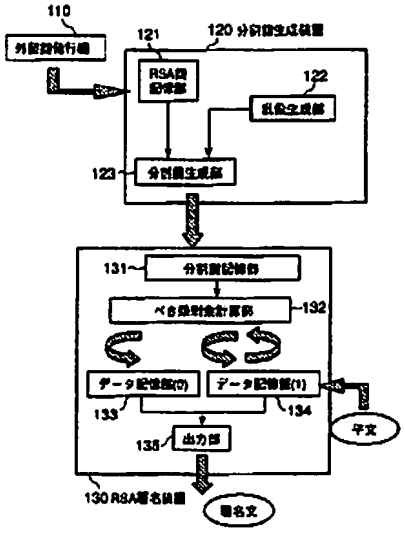
(21)出願番号　特願2002-160231(P2002-160231)
(22)出願日　平成14年5月31日(2002.5.31)

(71)出願人　000003078
株式会社東芝
東京都港区芝浦一丁目１番１号
(74)代理人　100058479
弁理士　鈴江　武彦
(74)代理人　100084618
弁理士　村松　貞男
(74)代理人　100068814
弁理士　坪井　淳
(74)代理人　100092196
弁理士　橋本　良郎
(74)代理人　100091351
弁理士　河野　哲
(74)代理人　100088683
弁理士　中村　誠

最終頁に続く

(54)【発明の名称】べき乗剰余計算装置、べき乗剰余計算方法及びプログラム

(57)【要約】
【課題】公開鍵暗号方式に対する電力解析攻撃による秘密鍵漏洩を防ぐことを可能にしたべき乗剰余計算装置を提供すること。
【解決手段】分割鍵生成部１２０は、公開鍵暗号方式の秘密鍵ｄからｋ個の分割秘密鍵を生成し、乱数によりｘ個のダミー鍵、ｋ個の１とｘ個の０からなる鍵インデックスを生成する。ＲＳＡ署名装置１３０は、データ記憶部（０），（１）に初期値として平文Ｍを書き込み、鍵インデックスが１ならばデータ記憶部（１）の値を入力として該当する１つの分割秘密鍵を用いてべき乗剰余算を行い、その結果をデータ記憶部（１）に格納し、０ならばデータ記憶部（０）の値を入力として該当する１つのダミー鍵を用いてべき乗剰余算を行い、その結果をデータ記憶部（０）に格納する。処理後のデータ記憶部（１）の値が署名文である。電力解析攻撃により分割秘密鍵が漏れても秘密鍵ｄを求めることは困難になる。
【選択図】　図１



【特許請求の範囲】
【請求項１】
公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置において、
前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づき分割して得られた、複数の分割秘密鍵を記憶する手段と、
前記複数の分割秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行う第１の計算手段とを備え、
前記第１の計算手段により行われる前記べき乗剰余算は、前記入力データ又は先行して行われた他のべき乗剰余算の計算結果を入力とするものであり、
前記入力データに対するべき乗剰余は、前記第１の計算手段により行われた特定の一つの前記べき乗剰余算の計算結果が示す値、又は特定の複数の前記べき乗剰余算の計算結果に基づいて得られる値であることを特徴とするべき乗剰余計算装置。
【請求項２】
前記秘密鍵をもとに前記複数の分割秘密鍵を生成する手段を更に備えたことを特徴とする請求項１に記載のべき乗剰余計算装置。
【請求項３】
複数のダミー鍵を記憶する手段と、
前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行う第２の計算手段と、
前記第１の計算手段と前記第２の計算手段とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を記憶する手段とを更に備えたことを特徴とする請求項１に記載のべき乗剰余計算装置。
【請求項４】
前記秘密鍵をもとに前記複数の分割秘密鍵を生成する手段と、
前記複数のダミー鍵を生成する手段と、
前記インデックス情報を生成する手段とを更に備えたことを特徴とする請求項３に記載のべき乗剰余計算装置。
【請求項５】
前記第１の計算手段及び前記第２の計算手段は、それぞれ、前記インデックス情報の当該回に対応するフラグによる指示に従って、該当する回のみにおいて前記所定のべき乗剰余算を順次行うことを特徴とする請求項３または４に記載のべき乗剰余計算装置。
【請求項６】
前記第２の計算手段における前記複数のダミー鍵の使用順序と、前記インデックス情報との少なくとも一方を、所定のタイミングで変更することを特徴とする請求項３ないし５のいずれか１項に記載のべき乗剰余計算装置。
【請求項７】
前記第１の計算手段における前記複数の分割秘密鍵の使用順序と、前記所定の分割方法との少なくとも一方を、所定のタイミングで変更することを特徴とする請求項１ないし６のいずれか１項に記載のべき乗剰余計算装置。
【請求項８】
公開鍵暗号方式の秘密鍵から派生した第１及び第２の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置において、
前記第１の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づき分割して得られた、複数の分割秘密鍵を記憶する手段と、
前記複数の第１の分割秘密鍵をそれぞれ用いた所定の第１のべき乗剰余算を順次行う第１の計算手段と、
前記第２の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づき分割して得られた、複数の分割秘密鍵を記憶する手段と、
前記複数の第２の分割秘密鍵をそれぞれ用いた所定の第２のべき乗剰余算を順次行う第２の計算手段と、

前記第１の計算手段と前記第２の計算手段とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を記憶するインデックス情報生成手段と、
前記第１の計算手段による最終結果と前記第２の計算手段による最終結果とに基づいて、前記入力データに対するべき乗剰余を求める手段とを備え、
前記第１の計算手段により行われる前記べき乗剰余算は、前記入力データ又は先行して行われた他の第１のべき乗剰余算の計算結果を入力とするものであり、
前記第２の計算手段により行われる前記べき乗剰余算は、前記入力データ又は先行して行われた他の第２のべき乗剰余算の計算結果を入力とするものであり、
前記第１の計算手段による最終結果は、前記第１の計算手段により行われた特定の一つの前記べき乗剰余算の計算結果が示す値、又は特定の複数の前記べき乗剰余算の計算結果に基づいて得られる値であり、
前記第２の計算手段による最終結果は、前記第２の計算手段により行われた特定の一つの前記べき乗剰余算の計算結果が示す値、又は特定の複数の前記べき乗剰余算の計算結果に基づいて得られる値であることを特徴とするべき乗剰余計算装置。
【請求項９】
前記第１の秘密鍵をもとに前記複数の第１の分割秘密鍵を生成する手段と、
前記第２の秘密鍵をもとに前記複数の第２の分割秘密鍵を生成する手段とを更に備えたことを特徴とする請求項８に記載のべき乗剰余計算装置。
【請求項１０】
前記第１の計算手段及び前記第２の計算手段は、それぞれ、前記インデックス情報の当該回に対応するフラグによる指示に従って、該当する回のみにおいて前記所定のべき乗剰余算を順次行うことを特徴とする請求項８または９に記載のべき乗剰余計算装置。
【請求項１１】
複数のダミー鍵を記憶する手段と、
前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行う第３の計算手段とを更に備え、
前記インデックス情報は、前記第１の計算手段乃至前記第３の計算手段のいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなることを特徴とする請求項８に記載のべき乗剰余計算装置。
【請求項１２】
前記第１の秘密鍵をもとに前記複数の第１の分割秘密鍵を生成する手段と、
前記第２の秘密鍵をもとに前記複数の第２の分割秘密鍵を生成する手段と、
前記複数のダミー鍵を生成する手段とを更に備えたことを特徴とする請求項１１に記載のべき乗剰余計算装置。
【請求項１３】
前記第１の計算手段乃至前記第３の計算手段は、それぞれ、前記インデックス情報の当該回に対応するフラグによる指示に従って、該当する回のみにおいて前記所定のべき乗剰余算を順次行うことを特徴とする請求項１１または１２に記載のべき乗剰余計算装置。
【請求項１４】
前記第１の計算手段における前記複数のダミー鍵の使用順序を所定のタイミングで変更することを特徴とする請求項１１ないし１３のいずれか１項に記載のべき乗剰余計算装置。
【請求項１５】
前記第１の計算手段における前記複数の第１の分割秘密鍵の使用順序と、前記第２の計算手段における前記複数の第２の分割秘密鍵の使用順序と、前記インデックス情報と、前記所定の分割方法との少なくとも一つを、所定のタイミングで変更することを特徴とする請求項８ないし１４のいずれか１項に記載のべき乗剰余計算装置。
【請求項１６】
公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、

前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の分割秘密鍵に分割するステップと、
前記複数の分割秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、
少なくとも一つの前記所定のべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とするべき乗剰余計算方法。
【請求項１７】
公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、
前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の分割秘密鍵に分割するステップと、
複数のダミー鍵を生成するステップと、
前記秘密鍵と前記ダミー鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成するステップと、
前記インデックス情報に従って、前記複数の秘密鍵及び前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、
少なくとも一つの前記秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とするべき乗剰余計算方法。
【請求項１８】
公開鍵暗号方式の秘密鍵から派生した第１及び第２の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、
前記第１の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第１の分割秘密鍵に分割するステップと、
前記第２の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第２の分割秘密鍵に分割するステップと、
前記第１の秘密鍵と前記第２の秘密鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成するステップと、
前記インデックス情報に従って、前記複数の第１の秘密鍵及び前記複数の第２の秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、
少なくとも一つの前記第１の秘密鍵を用いたべき乗剰余算の計算結果及び一つの前記第２の秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とするべき乗剰余計算方法。
【請求項１９】
公開鍵暗号方式の秘密鍵から派生した第１及び第２の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、
前記第１の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第１の分割秘密鍵に分割するステップと、
前記第２の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第２の分割秘密鍵に分割するステップと、
複数のダミー鍵を生成するステップと、
前記第１の秘密鍵と前記第２の秘密鍵と前記ダミー鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成するステップと、
前記インデックス情報に従って、前記複数の第１の秘密鍵及び前記複数の第２の秘密鍵並びに前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、
少なくとも一つの前記第１の秘密鍵を用いたべき乗剰余算の計算結果及び一つの前記第２の秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とするべき乗剰余計算方法。
【請求項２０】
公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置としてコンピュータを機能させるためのプログラムであって、

前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の分割秘密鍵に分割する機能と、
前記複数の分割秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行う機能と、
少なくとも一つの前記所定のべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求める機能とをコンピュータに実現させるためのプログラム。
【請求項２１】
公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置としてコンピュータを機能させるためのプログラムであって、
前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の分割秘密鍵に分割する機能と、
複数のダミー鍵を生成する機能と、
前記秘密鍵と前記ダミー鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成する機能と、
前記インデックス情報に従って、前記複数の秘密鍵及び前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行う機能と、
少なくとも一つの前記秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求める機能とをコンピュータに実現させるためのプログラム。
【請求項２２】
公開鍵暗号方式の秘密鍵から派生した第１及び第２の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置としてコンピュータを機能させるためのプログラムであって、
前記第１の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第１の分割秘密鍵に分割する機能と、
前記第２の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第２の分割秘密鍵に分割する機能と、
前記第１の秘密鍵と前記第２の秘密鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成する機能と、
前記インデックス情報に従って、前記複数の第１の秘密鍵及び前記複数の第２の秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行う機能と、
少なくとも一つの前記第１の秘密鍵を用いたべき乗剰余算の計算結果及び一つの前記第２の秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求める機能とをコンピュータに実現させるためのプログラム。
【請求項２３】
公開鍵暗号方式の秘密鍵から派生した第１及び第２の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置としてコンピュータを機能させるためのプログラムであって、
前記第１の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第１の分割秘密鍵に分割する機能と、
前記第２の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第２の分割秘密鍵に分割する機能と、
複数のダミー鍵を生成する機能と、
前記第１の秘密鍵と前記第２の秘密鍵と前記ダミー鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成する機能と、
前記インデックス情報に従って、前記複数の第１の秘密鍵及び前記複数の第２の秘密鍵並びに前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行う機能と、
少なくとも一つの前記第１の秘密鍵を用いたべき乗剰余算の計算結果及び一つの前記第２の秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求める機能とを有することを機能とをコンピュータに実現させるためのプログラム。
【発明の詳細な説明】
【０００１】

【発明の属する技術分野】
本発明は、公開鍵暗号方式の秘密鍵を用いてべき乗剰余計算を行うべき乗剰余計算装置、べき乗剰余計算方法及びプログラムに関する。
【０００２】
【従来の技術】
公開鍵暗号方式の中で、最も広く利用されているＲＳＡ暗号を例に挙げて説明する。
【０００３】
ＲＳＡ暗号では、最初に、２つの大きな素数ｐ，ｑを決定し、ｎ＝ｐ＊ｑ，　ｅｄ＝＝１（ｍｏｄ　ｌｃｍ（ｐ－１）（ｑ－１））を満たすｎ，ｅ，ｄを求めることによって、公開鍵（ｅ，ｎ）および秘密鍵（ｄ，ｐ，ｑ）となる鍵の生成を行う。なお、本明細書では、＝＝は、合同を表すものとする。
【０００４】
次に、例えばＲＳＡ署名を行う場合は、署名したい文Ｍに対して（秘密鍵を用いて）、Ｓ＝Ｍ＾ｄ　ｍｏｄ　ｎなる処理を行うことになる。
【０００５】
ところで、Ｓ＝Ｍ＾ｄ　ｍｏｄ　ｎをその通りに計算すると、ｄはｎと同程度に大きい（例えば５１２ビットや１０２４ビット程度）ため、べき乗に大変時間がかかってしまう。そこで、この計算には、通常、ｓｑｕａｒｅ－ｍｕｌｔｉｐｌｙ　ｍｅｔｈｏｄが使われる。
【０００６】
図１２のフローチャートを参照しながら、ｓｑｕａｒｅ－ｍｕｌｔｉｐｌｙ　ｍｅｔｈｏｄの流れを説明する。
【０００７】
ここで、例えば署名したい文Ｍ＝ｘおよび秘密鍵ｄ＝ｂとして、ｘ＾ｂ　ｍｏｄ　ｎを例にする。
【０００８】
べき指数ｂは、
ｂ［ｋ－１］，ｂ［ｋ－２］，…，ｂ［０］
と２進展開する（ただし、ｂ［ｋ－１］がＭＳＢ、ｂ［０］がＬＳＢである）。
【０００９】
初期値をｚ＝１，ｉ＝ｋ－１とする（ステップＳ１０１）。以降は、繰り返し処理となる。
【００１０】
すなわち、まず、ｚ＝ｚ＾２　ｍｏｄ　ｎなる処理をする（ステップＳ１０２）。
【００１１】
次いで、べき指数のビットｂ［ｉ］が１ならば（ステップＳ１０３）、ｚ＝ｚ＊ｘ　ｍｏｄ　ｎを計算し（ステップＳ１０４）、ｉをデクリメントする（ステップＳ１０５）。他方、べき指数のビットｂ［ｉ］が０ならば（ステップＳ１０３）、何もせずに、ｉをデクリメントする（ステップＳ１０５）。
【００１２】
そして、ｉ＜０でないならば（ステップＳ１０６）、ステップＳ１０２に戻り、次のループの処理を行う。
【００１３】
最終的に、ステップＳ１０６において、ｉ＜０となったならば、処理を終了する。このときのｚが、求める解すなわちｘ＾ｂ　ｍｏｄ　ｎである。
【００１４】
このようなアルゴリズムを利用することで、べき乗剰余算の処理時間を大幅に短くすることができる。
【００１５】
さらに、中国人剰余定理（Ｃｈｉｎｅｓｅ　Ｒｅｍａｉｄａｒ　Ｔｈｅｏｒｅｍ（以下、

ＣＲＴと略す））を利用して処理時間を短縮する方法もある。これは、例えばＲＳＡの処理の場合では、べき乗剰余の法をｐ、ｑに分解して、それぞれを法とした計算処理を行い、最後に結果を合成するという方法である。

【００１６】

まず、Ｓ＝Ｍ＾ｄ　ｍｏｄ　ｎを、

Ｓｐ＝Ｍｐ＾ｄｐ　ｍｏｄ　ｐ

Ｓｑ＝Ｍｑ＾ｄｑ　ｍｏｄ　ｑ

の２つに分解する。

ここで、

Ｍｐ＝Ｍ　ｍｏｄ　ｐ

Ｍｑ＝Ｍ　ｍｏｄ　ｑ

ｄｐ＝ｄ　ｍｏｄ　（ｐ－１）

ｄｑ＝ｄ　ｍｏｄ　（ｑ－１）

である。

【００１７】

しかして、ＳｐとＳｑをそれぞれ計算した後に、

Ｓ＝｛Ｓｐ＊ｑ＊（ｑ＾－１　ｍｏｄ　ｐ）＋Ｓｑ＊ｐ＊（ｐ＾－１　ｍｏｄ　ｑ）｝

ｍｏｄ　ｎ

という処理をして合成することによって、Ｓ＝Ｍ＾ｄ　ｍｏｄ　ｎの結果を得ることができる。

【００１８】

この場合においても、Ｓｐ＝Ｍｐ＾ｄｐ　ｍｏｄ　ｐの計算と、Ｓｑ＝Ｍｑ＾ｄｑ　ｍｏｄ　ｑの計算のそれぞれについて、ｓｑｕａｒｅ－ｍｕｌｔｉｐｌｙ　ｍｅｔｈｏｄが用いられる。

【００１９】

なお、上記では、ＲＳＡ署名を例にとって説明したが、ＲＳＡ署名の検証、ＲＳＡの暗号化、ＲＳＡの復号など、べき乗剰余算を行う他の暗号処理についても同様である。

【００２０】

【発明が解決しようとする課題】

ところで、上記のような従来の方法でＲＳＡ暗号を実装すると、ｓｑｕａｒｅ－ｍｕｌｔｉｐｌｙ　ｍｅｔｈｏｄの際に、べき指数の２進展開で、ｂ［ｉ］が０のときと１のときで処理が違うために（図１２のＳ１０３での分岐を参照）、暗号処理中の装置の電力を詳細に測定することで、消費電力波形に１のときの処理と０のときの処理との違いが観測され、結果として暗号の秘密鍵の情報が漏洩してしまうという問題点がある。例えば、通常のＲＳＡの処理では秘密鍵ｄの値が、中国人剰余定理を用いたＲＳＡの処理では秘密鍵ｄに対応するｄｐ，ｄｑの値が、消費電力波形から分かってしまう。このような攻撃法は、電力解析攻撃と呼ばれる。

【００２１】

本発明は、上記事情を考慮してなされたもので、公開鍵暗号方式に対する電力解析攻撃による秘密鍵漏洩を防ぐことを可能にしたべき乗剰余計算装置、べき乗剰余計算方法及びプログラムを提供することを目的とする。

【００２２】

【課題を解決するための手段】

本発明は、公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置において、前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づき分割して得られた、複数の分割秘密鍵を記憶する手段と、前記複数の分割秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行う第１の計算手段とを備え、前記第１の計算手段により行われる前記べき乗剰余算は、前記入力データ又は先行して行われた他のべき乗剰余算の計算結果を入力とするものであり、前記入力データに対するべき乗剰余は、前記第１の計算手段により行われた特定の一つの前記べき乗

剰余算の計算結果が示す値、又は特定の複数の前記べき乗剰余算の計算結果に基づいて得られる値であることを特徴とする。

【００２３】

また、本発明は、公開鍵暗号方式の秘密鍵から派生した第１及び第２の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置において、前記第１の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づき分割して得られた、複数の分割秘密鍵を記憶する手段と、前記複数の第１の分割秘密鍵をそれぞれ用いた所定の第１のべき乗剰余算を順次行う第１の計算手段と、前記第２の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づき分割して得られた、複数の分割秘密鍵を記憶する手段と、前記複数の第２の分割秘密鍵をそれぞれ用いた所定の第２のべき乗剰余算を順次行う第２の計算手段と、前記第１の計算手段と前記第２の計算手段とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を記憶するインデックス情報生成手段と、前記第１の計算手段による最終結果と前記第２の計算手段による最終結果とに基づいて、前記入力データに対するべき乗剰余を求める手段とを備え、前記第１の計算手段により行われる前記べき乗剰余算は、前記入力データ又は先行して行われた他の第１のべき乗剰余算の計算結果を入力とするものであり、前記第２の計算手段により行われる前記べき乗剰余算は、前記入力データ又は先行して行われた他の第２のべき乗剰余算の計算結果を入力とするものであり、前記第１の計算手段による最終結果は、前記第１の計算手段により行われた特定の一つの前記べき乗剰余算の計算結果が示す値、又は特定の複数の前記べき乗剰余算の計算結果に基づいて得られる値であり、前記第２の計算手段による最終結果は、前記第２の計算手段により行われた特定の一つの前記べき乗剰余算の計算結果が示す値、又は特定の複数の前記べき乗剰余算の計算結果に基づいて得られる値であることを特徴とする。

【００２４】

また、本発明は、公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の分割秘密鍵に分割するステップと、前記複数の分割秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、少なくとも一つの前記所定のべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とする。

【００２５】

また、本発明は、公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の分割秘密鍵に分割するステップと、複数のダミー鍵を生成するステップと、前記秘密鍵と前記ダミー鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成するステップと、前記インデックス情報に従って、前記複数の秘密鍵及び前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、少なくとも一つの前記秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とする。

【００２６】

また、本発明は、公開鍵暗号方式の秘密鍵から派生した第１及び第２の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、前記第１の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第１の分割秘密鍵に分割するステップと、前記第２の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第２の分割秘密鍵に分割するステップと、前記第１の秘密鍵と前記第２の秘密鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成するステップと、前記インデックス情報に従って、前記複数の第１の秘密鍵及び前記複数の第２の秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、少なくと

も一つの前記第１の秘密鍵を用いたべき乗剰余算の計算結果及び一つの前記第２の秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とする。

【００２７】

また、本発明は、公開鍵暗号方式の秘密鍵から派生した第１及び第２の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、前記第１の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第１の分割秘密鍵に分割するステップと、前記第２の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第２の分割秘密鍵に分割するステップと、複数のダミー鍵を生成するステップと、前記第１の秘密鍵と前記第２の秘密鍵と前記ダミー鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成するステップと、前記インデックス情報に従って、前記複数の第１の秘密鍵及び前記複数の第２の秘密鍵並びに前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、少なくとも一つの前記第１の秘密鍵を用いたべき乗剰余算の計算結果及び一つの前記第２の秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とする。

【００２８】

なお、装置に係る本発明は方法に係る発明としても成立し、方法に係る本発明は装置に係る発明としても成立する。

また、装置または方法に係る本発明は、コンピュータに当該発明に相当する手順を実行させるための（あるいはコンピュータを当該発明に相当する手段として機能させるための、あるいはコンピュータに当該発明に相当する機能を実現させるための）プログラムとしても成立し、該プログラムを記録したコンピュータ読取り可能な記録媒体としても成立する。

【００２９】

本発明では、公開鍵暗号方式の秘密鍵（例えば、中国人剰余定理が用いられる場合には該秘密鍵から各法ｐ，ｑに対応する２つの秘密鍵ｄｐ，ｄｑのそれぞれ）をさらに分割した分割秘密鍵を用いて処理をするため、分割された一つ一つの分割秘密鍵は消費電力から分かってしまっても、それらから秘密鍵を求めることが困難である。本発明によれば、計算量的安全性によって秘密鍵漏洩を防ぐことが可能になる。さらに、秘密鍵の分割の仕方として、加法、減法、乗法若しくは除法またはそれらを組み合わせたものというように、あらゆる組み合わせで構成することができ、また、分割数や分割する鍵長を可変にすることや、ダミー鍵を挿入するなど、状況に応じてセキュリティレベルを変えることができる。

【００３０】

【発明の実施の形態】

以下、図面を参照しながら発明の実施の形態を説明する。

【００３１】

以下では、公開鍵暗号方式のうちＲＳＡ暗号の場合を例にとって説明する。また、以下では、秘密鍵を用いたべき乗剰余計算としてＲＳＡ暗号の署名生成の場合を例にとって説明する。

【００３２】

（第１の実施形態）

まず、本発明の第１の実施形態として、中国人剰余定理を使用しない場合の実施形態について説明する。

【００３３】

図１に、本実施形態に係る暗号処理システムの構成例を示す。図１は、分割鍵生成装置とＲＳＡ署名装置（べき乗剰余計算装置）とが分離している分離型の構成をとる場合の一例である。

【００３４】

図１に示されるように、本暗号処理システムは、外部鍵発行機１１０、分割鍵生成装置１２０、ＲＳＡ署名装置１３０を備えている。分割鍵生成装置１２０は、ＲＳＡ鍵記憶部１２１、乱数生成部１２２、分割鍵生成部１２３を含み、ＲＳＡ署名装置１３０は、分割鍵記憶部１３１、べき乗剰余計算部１３２、第１のデータ記憶部（以下、データ記憶部（０）と記述する）１３３、第２のデータ記憶部（以下、データ記憶部（１）と記述する）１３４、出力部１３５を含む。

【００３５】

外部鍵発行機１１０は、ＲＳＡの公開鍵（ｅ，ｎ）と秘密鍵（ｄ，ｐ，ｑ）を生成し、これらを分割鍵生成装置１２０内のＲＳＡ鍵記憶部１２１に供給する。ただし、ｐ，ｑはそれぞれ大きな素数で、ｎ＝ｐ＊ｑ，　ｅｄ＝＝１　（ｍｏｄ　ｌｃｍ（ｐ－１）（ｑ－１））を満たす。

【００３６】

分割鍵生成装置１２０は、分割秘密鍵等を生成してＲＳＡ署名装置１３０に供給し、ＲＳＡ署名装置１３０は、供給された分割秘密鍵等を用いて平文Ｍに対する署名文Ｍ＾ｄ　ｍｏｄ　ｎを求めるものである。

【００３７】

まず、分割鍵生成装置１２０について説明する。

【００３８】

ＲＳＡ鍵記憶部１２１は、外部鍵発行機１１０から供給されるＲＳＡの公開鍵（ｅ，ｎ）と秘密鍵（ｄ，ｐ，ｑ）を保持し、必要に応じて分割鍵生成部１２３に公開鍵（ｅ，ｎ）と秘密鍵（ｄ，ｐ，ｑ）を供給する。

【００３９】

乱数生成部１２２は、任意のビット長の鍵およびビット列を任意の数だけ乱数により生成し、それらを分割鍵生成部１２３に供給する。

【００４０】

分割鍵生成部１２３は、乱数生成部１２２から供給される分割秘密鍵の構成成分と、ＲＳＡ鍵記憶部１２１から供給される秘密鍵（ｄ，ｐ，ｑ）、公開鍵（ｅ，ｎ）とを用いて、分割秘密鍵を生成する。

【００４１】

以下、中国人剰余定理を使用しない場合の分割秘密鍵生成手順について説明する。

【００４２】

図２に、この場合の処理手順の一例を示す。

【００４３】

ここでは、分割秘密鍵生成の一例として、秘密鍵ｄを、乗法のみにより、ｋ個に分割するものとする。すなわち、

ｄ＝ｄ［０］＊ｄ［１］＊…＊ｄ［ｋ－１］　ｍｏｄ　（ｌｃｍ（ｐ－１，ｑ－１））

となるように、ｄをｄ［０］，…，ｄ［ｋ－１］に分割するものとする。

【００４４】

まず、乱数生成部１２２は、ｓビットの乱数をｋ個生成し、それぞれ、分割秘密鍵ｄ［１］～ｄ［ｋ－１］へ代入する（ステップＳ１）。なお、ｓは、秘密鍵ｄのビット長より小さいビット数（秘密鍵ｄが例えば５１２ビットあるいは１０２４ビット程度の場合に分割秘密鍵ｄ［１］～ｄ［ｋ－１］がそれぞれ例えば８ビット程度）にすると好ましい。

【００４５】

また、乱数生成部１２２は、ｓビットの乱数をｘ個生成し、それぞれ、ダミー鍵ｄｕ［０］～ｄｕ［ｘ－１］へ代入する（ステップＳ２）。

【００４６】

また、乱数生成部１２２は、ｋ＋ｘ－１ビット長で且つハミング重みをｋ－１とする鍵インデックスｔ［ｉ］　（１≦ｉ≦ｋ＋ｘ－１）をランダムに生成する（ステップＳ３）。

【００４７】

次いで、分割鍵生成部１２３は、ｔ［ｉ］＝１となる鍵（つまり、分割秘密鍵ｄ［ｊ］（

１≦ｊ≦ｋ－１））を掛け合わせて、
$D_0 = d[1] * d[2] * \cdots * d[k-1]$
とし、ＲＳＡ鍵記憶部１２１から供給される秘密鍵ｄを用いて、
$d[0] = d * D_0^{-1} \mod (lcm(p-1, q-1))$
とすることで、残り１つの分割秘密鍵ｄ［０］を求める（ステップＳ４）。
また、ｔ［０］＝１とする（ステップＳ４）。

【００４８】
以上で、
分割秘密鍵ｄ［ｊ］（０≦ｊ≦ｋ－１）、
ダミー鍵ｄｕ［ｕ］（０≦ｕ≦ｘ－１）、
鍵インデックスｔ［ｉ］（０≦ｉ≦ｋ＋ｘ－１）
が完成する。

【００４９】
そして、分割鍵生成部１２３は、
ｄ［ｊ］（０≦ｊ≦ｋ－１）、
ｄｕ［ｕ］（０≦ｕ≦ｘ－１）、
ｔ［ｉ］（０≦ｉ≦ｋ＋ｘ－１）、
公開鍵（ｅ，ｎ）、
秘密鍵（ｄ，ｐ，ｑ）
をＲＳＡ署名装置１３０へ出力する（ステップＳ５）。

【００５０】
なお、図２の手順においては、ｄ［１］，ｄ［２］，…，ｄ［ｋ－１］を全て生成した後にｄ［０］を求める必要があるが、それ以外は、ｋ個の分割秘密鍵とｘ個のダミー鍵と鍵インデックスをどのような順番で求めても構わない。

【００５１】
次に、ＲＳＡ署名装置１３０について説明する。

【００５２】
分割鍵記憶部１３１は、分割鍵生成装置１２０内の分割鍵生成部１２３から供給される分割秘密鍵、ダミー鍵、鍵インデックス、公開鍵（ｅ，ｎ）、秘密鍵（ｄ，ｐ，ｑ）を保持し、必要に応じてそれぞれをべき乗剰余計算部１３２に供給する。

【００５３】
データ記憶部（０）１３３には、ダミー鍵を使用したべき乗剰余計算の計算結果（すなわちダミーの計算結果）Ｓ’が格納され、データ記憶部（１）１３４には、分割秘密鍵を使用したべき乗剰余計算の計算結果（すなわち正規の計算結果）Ｓが格納される。

【００５４】
べき乗剰余計算部１３２は、データ記憶部（１）１３４から前回までの計算結果Ｓ（ただし、初回は初期値Ｓ＝Ｍとなる）を呼び出して、分割鍵記憶部１３１から供給される鍵インデックス値ｔ［ｉ］が１ならば、分割秘密鍵ｄ［ｊ］を使ってＳ＝Ｓ＾ｄ［ｊ］ ｍｏｄ ｎの計算を行い、その結果Ｓをデータ記憶部（１）１３４に格納し、他方、鍵インデックス値ｔ［ｉ］が０ならば、ダミー鍵ｄｕ［ｕ］を使ってＳ’＝Ｓ＾ｄｕ［ｕ］ ｍｏｄ ｎの計算をし、その結果Ｓ’をデータ記憶部（０）１３３に格納する。なお、鍵インデックス値ｔ［ｉ］＝０の場合に、Ｓ’＝Ｓ＾ｄｕ［ｕ］ ｍｏｄ ｎ以外の計算を行う構成も可能である。

【００５５】
なお、ここでは、鍵インデックス値ｔ［ｉ］＝１の場合に、分割秘密鍵によるべき乗剰余計算を行い、鍵インデックス値ｔ［ｉ］＝０の場合に、ダミー鍵によるべき乗剰余計算を行うようにしているが、これに限定されるものではなく、例えば、鍵インデックス値ｔ［ｉ］＝０の場合に、分割秘密鍵によるべき乗剰余計算を行い、鍵インデックス値ｔ［ｉ］＝１の場合に、ダミー鍵によるべき乗剰余計算を行うようにしてもよい。

【００５６】

出力部１３５は、データ記憶部（１）１３４の最終結果Ｓ（＝Ｍ＾ｄ　ｍｏｄｎとなる）を参照し、これを最終署名結果として出力する。

【００５７】

以下、中国人剰余定理を使用しない場合の分割秘密鍵を用いたＲＳＡ署名手順について説明する。

【００５８】

図３に、この場合の処理手順の一例を示す。

【００５９】

なお、ここでは、分割秘密鍵ｄ［ｊ］は、ｊの値の小さい方から逐次使用し、ダミー鍵ｄｕ［ｕ］は、ｕの値の小さい方から逐次使用するものとした例を示している。

【００６０】

最初に、データ記憶部（０）１３３およびデータ記憶部（１）１３４にそれぞれ初期値として平文Ｍを書き込み、また、ｉ＝０、ｊ＝０、ｕ＝０とする（ステップＳ１１）。なお、データ記憶部（０）１３３については、平文Ｍ以外の値が初期値となっても構わない。

【００６１】

以降は、ｋ＋ｘ回の繰り返し処理となる。

【００６２】

まず、べき乗剰余計算部１３２は、鍵インデックス値ｔ［ｉ］を参照し、ｔ［ｉ］＝１ならば（ステップＳ１２）、データ記憶部（１）１３４からＳを読み出し、分割秘密鍵ｄ［ｊ］を使って、Ｓ＝Ｓ＾ｄ［ｊ］　ｍｏｄ　ｎの計算を行い、その結果Ｓをデータ記憶部（１）１３４に格納し（ステップＳ１３）、ｊおよびｉをそれぞれ１ずつインクリメントする（ステップＳ１４，Ｓ１７）。

【００６３】

他方、鍵インデックス値ｔ［ｉ］が０ならば（ステップＳ１２）、データ記憶部（１）１３４からＳを読み出し、ダミー鍵ｄｕ［ｕ］を使って、Ｓ’＝Ｓ＾ｄｕ［ｕ］　ｍｏｄｎの計算を行い、その結果Ｓ’をデータ記憶部（０）１３３に格納し（ステップＳ１５）、ｕおよびｉをそれぞれ１ずつインクリメントする（ステップＳ１６，Ｓ１７）。

【００６４】

上記の２系統のうちの鍵インデックス値ｔ［ｉ］に応じたいずれかが行われた後に、ｉの値を参照し、ｉの値が分割秘密鍵とダミー鍵の総数ｋ＋ｘを超えていないなら（ステップＳ１８）、ステップＳ１２に戻り、次のループの処理を行う。

【００６５】

最終的に、ステップＳ１８において、ｉ＞（ｋ＋ｘ）となったならば、処理を終了する。このときにデータ記憶部（１）１３４に格納されているＳが、求める署名文すなわちＭ＾ｄ　ｍｏｄ　ｎである。

【００６６】

以上のように処理を行うことで、分割秘密鍵によるべき乗剰余計算とダミー鍵によるべき乗剰余計算とをランダムな順序に実行させて、電力解析攻撃を困難にさせ、かつ、ダミー鍵の影響を残さずに、正しい最終署名結果を得ることができる。

【００６７】

ここで、具体例で説明すると、例えば、ｋ＝８、ｘ＝４、鍵インデックス＝“１０１１０１１１０１０１”の場合、

（１回目）Ｓ　＝Ｍ＾ｄ［０］　ｍｏｄ　ｎ
（２回目）Ｓ’＝Ｓ＾ｄｕ［０］　ｍｏｄ　ｎ
（３回目）Ｓ　＝Ｓ＾ｄ［１］　ｍｏｄ　ｎ
（４回目）Ｓ　＝Ｓ＾ｄ［２］　ｍｏｄ　ｎ
（５回目）Ｓ’＝Ｓ＾ｄｕ［１］　ｍｏｄ　ｎ
（６回目）Ｓ　＝Ｓ＾ｄ［３］　ｍｏｄ　ｎ
（７回目）Ｓ　＝Ｓ＾ｄ［４］　ｍｏｄ　ｎ
（８回目）Ｓ　＝Ｓ＾ｄ［５］　ｍｏｄ　ｎ

（９回目）Ｓ' ＝Ｓ＾ｄｕ［２］　ｍｏｄ　ｎ
（１０回目）Ｓ　＝Ｓ＾ｄ［６］　ｍｏｄ　ｎ
（１１回目）Ｓ' ＝Ｓ＾ｄｕ［３］　ｍｏｄ　ｎ
（１２回目）Ｓ　＝Ｓ＾ｄ［７］　ｍｏｄ　ｎ
のように各ループ処理が行われ、
最終的に得られたＳ
＝（（（（（（（Ｍ＾ｄ［０］　ｍｏｄ　ｎ）＾ｄ［１］　ｍｏｄ　ｎ）＾ｄ［２］　ｍｏｄ　ｎ）＾ｄ［３］　ｍｏｄ　ｎ）＾ｄ［４］　ｍｏｄｎ）＾ｄ［５］　ｍｏｄ　ｎ）＾ｄ［６］　ｍｏｄ　ｎ）＾ｄ［７］　ｍｏｄ　ｎ
＝Ｍ＾ｄ［０］＾ｄ［１］＾ｄ［２］＾ｄ［３］＾ｄ［４］＾ｄ［５］＾ｄ［６］＾ｄ［７］　ｍｏｄ　ｎ
＝Ｍ＾（ｄ［０］＊ｄ［１］＊ｄ［２］＊ｄ［３］＊ｄ［４］＊ｄ［５］＊ｄ［６］＊ｄ［７］）　ｍｏｄ　ｎ
＝Ｍ＾ｄ　ｍｏｄ　ｎ
となる。

【００６８】
このように本実施形態においては、電力解析攻撃によってＲＳＡ署名処理中の消費電力波形から、たとえ分割秘密鍵とダミー鍵の全ての値が分かったとしても、ダミー鍵と分割秘密鍵との区別が特定されなければ、秘密鍵ｄを特定することができない。したがって、秘密鍵を求めるために必要な総当り数（測定される波形の個数ｘ＋ｋから分割秘密鍵の個数ｋを選び出す組合せ）を大きくすることで、計算量的に攻撃を困難にすることができる。これまで説明した構成例では、鍵の分割数ｋを多くすればするほど、セキュリティ強度が上がる。また、ダミー鍵の数を多くすればするほど、セキュリティ強度が上がる。

【００６９】
以下では、本実施形態の変形例について説明する。

【００７０】
上記では、分割秘密鍵ｄ［ｊ］は、ｊの値の小さい方から逐次使用し、ダミー鍵ｄｕ［ｕ］は、ｕの値の小さい方から逐次使用するものとしたが、もちろん、これらとは異なる順番で鍵を使用していってもよい。

【００７１】
また、上記では、分割秘密鍵ｄ［ｊ］の使用順序およびダミー鍵ｄｕ［ｕ］の使用順序は固定のものであったが、分割秘密鍵の使用順序とダミー鍵の使用順序の一方又は双方を、所定のタイミングで（例えば、毎回（本例では、署名文を生成するごとに））、ランダムに変えるようにしてもよい。例えば、毎回、乱数に基づいて全鍵の使用順序を決定するようにしてもよいし、また、例えば、予め鍵の使用順序のパターンを複数パターン用意しておき、それらパターンのうちから使用するものを乱数に基づいて決定するようにしてもよい。このように鍵の使用順序をランダムにすることによって、さらにセキュリティ強度を上げることができる。

【００７２】
また、上記では、秘密鍵ｄの分割方法として、乗法のみのものを使用したが、他の分割方法、例えば、加法のみの分割方法（例えば、ｄ＝ｄ［０］＋ｄ［１］＋ｄ［２］＋ｄ［３］＋…）、減法のみの分割方法（例えば、ｄ＝ｄ［０］－ｄ［１］－ｄ［２］－ｄ［３］－…）、除法のみの分割方法（例えば、ｄ＝ｄ［０］／ｄ［１］／ｄ［２］／ｄ［３］／…）も可能であり、また、加法・減法・乗法・除法の適当な組合せも可能である（例えば、ｄ＝ｄ［０］＊ｄ［１］＋ｄ［２］＊ｄ［３］＋…）。

【００７３】
また、上記では、秘密鍵ｄの分割方法は特定のものに固定されていたが、例えば、装置（あるいはユーザ）毎に秘密鍵の分割方法を変える（同一の装置（あるいはユーザ）では秘密鍵の分割方法は固定する）ようにしてもよいし、また、例えば、同一の装置（あるいはユーザ）についても、所定のタイミングで（例えば、毎回）、鍵の分割方法を変えるよう

にしてもよい。このように秘密鍵の分割方法をランダムにすることによって、さらにセキュリティ強度を上げることができる。

【００７４】

なお、分割方法を変えるにあたっては、分割方法自体を異なるものに変えてもよい。また、分割方法自体を異なるものに変えるのではなく、例えば、ｊ番目の分割鍵ｄ［ｊ］に対する演算記号が加減乗除のいずれになるかかが予め定められている特定の分割方法を用いるものとして、この特定の分割方法は変えずに、秘密鍵の分割数のみを変えるようにしてもよい。例えば、上記の乗算のみの分割方法の例では、各分割鍵に対する演算記号は必ず乗算記号になるので、分割方法としては乗法のみを用いた分割方法を用いるものとして、秘密鍵ｄの分割数ｋを可変とする場合が該当する。これらのようにすることによって、さらにセキュリティ強度を上げることができる。

【００７５】

ここで、乗法以外のみの分割方法の一例として、加法のみの分割方法の場合について、前述した乗法に関する構成例を修正する部分について説明する。

【００７６】

まず、秘密鍵ｄを加法によりｋ分割すると、分割鍵は次のようになる。

$d = d[0] + d[1] + \cdots + d[k-1] \mod (lcm(p-1, q-1))$

$D_0 = d[1] + d[2] + \cdots + d[k-1]$

$d[0] = d - D_0 \mod (lcm(p-1, q-1))$

なお、ｄ［１］，ｄ［２］…ｄ［ｋ－１］は、乗法に関する構成例と同様、ｓビットの乱数を代入すればよい。

【００７７】

また、図３の処理手順例においては、ステップＳ１１でＳの初期値を１とし、ステップＳ１３のＳ＝Ｓ＾ｄ［ｉ］ ｍｏｄ ｎを、Ｓ＝Ｓ＊Ｍ＾ｄ［ｉ］ ｍｏｄ ｎとすればよい。なお、ステップＳ１５については、Ｓ＝Ｓ＊Ｍ＾ｄ［ｉ］ ｍｏｄ ｎに対応させて、Ｓ’＝Ｓ＊Ｍ＾ｄｕ［ｕ］ ｍｏｄ ｎとすればよい。

【００７８】

ここで、具体例で説明すると、例えば、ｋ＝８、ｘ＝４、鍵インデックス＝“１０１１０１１１０１０１”の場合、

（１回目）Ｓ ＝１＊Ｍ＾ｄ［０］ ｍｏｄ ｎ

（２回目）Ｓ’＝Ｓ＊Ｍ＾ｄｕ［０］ ｍｏｄ ｎ

（３回目）Ｓ ＝Ｓ＊Ｍ＾ｄ［１］ ｍｏｄ ｎ

（４回目）Ｓ ＝Ｓ＊Ｍ＾ｄ［２］ ｍｏｄ ｎ

（５回目）Ｓ’＝Ｓ＊Ｍ＾ｄｕ［１］ ｍｏｄ ｎ

（６回目）Ｓ ＝Ｓ＊Ｍ＾ｄ［３］ ｍｏｄ ｎ

（７回目）Ｓ ＝Ｓ＊Ｍ＾ｄ［４］ ｍｏｄ ｎ

（８回目）Ｓ ＝Ｓ＊Ｍ＾ｄ［５］ ｍｏｄ ｎ

（９回目）Ｓ’＝Ｓ＊Ｍ＾ｄｕ［２］ ｍｏｄ ｎ

（１０回目）Ｓ ＝Ｓ＊Ｍ＾ｄ［６］ ｍｏｄ ｎ

（１１回目）Ｓ’＝Ｓ＊Ｍ＾ｄｕ［３］ ｍｏｄ ｎ

（１２回目）Ｓ ＝Ｓ＊Ｍ＾ｄ［７］ ｍｏｄ ｎ

のように各ループ処理が行われ、

最終的に得られたS

$$=(((((((1*M\hat{}d[0] \bmod n)*M\hat{}d[1] \bmod n)*M\hat{}d[2] \bmod n)$$
$$*M\hat{}d[3] \bmod n)*M\hat{}d[4] \bmod n)*M\hat{}d[5] \bmod n)*M\hat{}d[6] \bmod n)$$
$$*M\hat{}d[7] \bmod n$$
$$=M\hat{}d[0]*M\hat{}d[1]*M\hat{}d[2]*M\hat{}d[3]*M\hat{}d[4]*M\hat{}d[5]*M\hat{}d[6]*M\hat{}d[7] \bmod n$$

$$=M\hat{}(d[0]+d[1]+d[2]+d[3]+d[4]+d[5]+d[6]+d[7]) \bmod n$$
$$=M\hat{}d \bmod n$$

となる。

【００７９】

また、他の分割方法の例として、乗算及び加法による分割方法の場合について、前述した乗法に関する構成例を修正する部分について説明する。

【００８０】

まず、秘密鍵ｄを乗算及び加法によりｋ分割した一例を次に示す（ただし、加法はｄ［ｋ／２－１］とｄ［ｋ／２］の間の１つのみとする）。

$d=d[0]*\cdots*d[k/2-1]+d[k/2]*\cdots*d[k-1] \bmod (lcm(p-1, q-1))$

$D_0=d[1]*\cdots*d[k/2-1]$

$D_1=d[k/2]*\cdots*d[k-1]$

$d[0]=(d-D_1)*D_0\hat{}-1 \bmod (lcm(p-1, q-1))$

なお、ｄ［１］，ｄ［２］…ｄ［ｋ－１］は、乗法に関する構成例と同様、ｓビットの乱数を代入すればよい。

【００８１】

また、図３の処理手順例においては、Ｓ１とＳ２の２つのパラメータを用意し、ステップＳ１１でＳ１およびＳ２の初期値をそれぞれＭとし、ステップＳ１３を、ｊ＝０～ｋ／２－１について、Ｓ１＝Ｓ１＾ｄ［ｉ］ ｍｏｄ ｎ、ｊ＝ｋ／２～ｋ－２について、Ｓ２＝Ｓ２＾ｄ［ｉ］ ｍｏｄ ｎ、ｊ＝ｋ－１について、Ｓ２＝Ｓ２＾ｄ［ｉ］ ｍｏｄｎを行った後に、Ｓ＝Ｓ１＊Ｓ２ ｍｏｄｎとすればよい。なお、ステップＳ１５については、例えば、Ｓ’＝Ｓ１＾ｄｕ［ｕ］ ｍｏｄ ｎあるいはＳ’＝Ｓ２＾ｄｕ［ｕ］ｍｏｄ ｎとすればよいし、また、例えば、ｕの値に応じて、Ｓ’＝Ｓ１＾ｄｕ［ｕ］ｍｏｄ ｎとＳ’＝Ｓ２＾ｄｕ［ｕ］ ｍｏｄ ｎのいずれを行うかを選択するようにしてもよい。

【００８２】

ここで、具体例で説明すると、例えば、ｋ＝８、ｘ＝４、鍵インデックス＝“１０１１０１１１０１０１”の場合、

（１回目）Ｓ１＝Ｍ＾ｄ［０］ ｍｏｄ ｎ
（２回目）Ｓ’＝Ｓ１＾ｄｕ［０］ ｍｏｄ ｎ
（３回目）Ｓ１＝Ｓ１＾ｄ［１］ ｍｏｄ ｎ
（４回目）Ｓ１＝Ｓ１＾ｄ［２］ ｍｏｄ ｎ
（５回目）Ｓ’＝Ｓ１＾ｄｕ［１］ ｍｏｄ ｎ
（６回目）Ｓ１＝Ｓ１＾ｄ［３］ ｍｏｄ ｎ
（７回目）Ｓ２＝Ｍ＾ｄ［４］ ｍｏｄ ｎ
（８回目）Ｓ２＝Ｓ２＾ｄ［５］ ｍｏｄ ｎ
（９回目）Ｓ’＝Ｓ２＾ｄｕ［２］ ｍｏｄ ｎ
（１０回目）Ｓ２＝Ｓ２＾ｄ［６］ ｍｏｄ ｎ
（１１回目）Ｓ’＝Ｓ２＾ｄｕ［３］ ｍｏｄ ｎ

（１２回目）Ｓ２＝Ｓ２＾ｄ［７］　ｍｏｄ　ｎ
のように各ループ処理が行われ、
最終的に得られたＳ
＝Ｓ１＊Ｓ２　ｍｏｄ　ｎ
＝((((Ｍ＾ｄ[0] mod n)＾ｄ[1] mod n)＾ｄ[2] mod n)＾ｄ[3] mod n)＊
((((Ｍ＾ｄ[4] mod n)＾ｄ[5] mod n)＾ｄ[6] mod n)＾ｄ[7] mod n)　ｍｏｄ
ｎ
＝Ｍ＾ｄ[0]*ｄ[1]*ｄ[2]*ｄ[3]*Ｍ*ｄ[4]*ｄ[5]*ｄ[6]*ｄ[7]　ｍｏｄ　ｎ
＝Ｍ＾（ｄ[0]*ｄ[1]*ｄ[2]*ｄ[3]+ｄ[4]*ｄ[5]*ｄ[6]*ｄ[7]）　ｍｏｄ　ｎ
＝Ｍ＾ｄ　ｍｏｄ　ｎ
となる。

【００８３】
さらに、他の分割方法も可能であり、分割方法に応じて、ｄ［０］や、図３の処理手順例のステップＳ１３の処理が定義される。ただし、分割方法によっては、ｄ［０］の値を求める過程で逆元計算（例えば、D_0＾－１）が必要になる場合に、その逆元計算において逆元が存在しないために、ｄ［０］の値を求めることができないときは、その分割方法を使用することはできない。したがって、逆元計算を要する分割方法においては、ｄ［１］，ｄ［２］…ｄ［ｋ－１］からｄ［０］を求めるにあたって、必要な逆元が存在するかどうかをチェックして、存在しなければ、もう一度乱数を発生させて、ｄ［１］，ｄ［２］…ｄ［ｋ－１］（の全部又は一部）を求め直すようにしてもよい。また、複数の分割方法を用意しておき、必要な逆元が存在しなければ、分割方法自体を変えるようにしてもよい。

【００８４】
ところで、図１は、分割鍵生成装置とＲＳＡ署名装置とが分離している分離型の構成をとる場合の一例であったが、分割鍵生成装置とＲＳＡ署名装置が一体化している一体型の構成をとることも可能である。図４に、この場合の暗号処理システムの構成例を示す。図４に示されるように、本暗号処理システムは、図１の外部鍵発行機１１０と同様の外部鍵発行機３１０と、図１の分割鍵生成装置１２０の構成要素とＲＳＡ署名装置１３０の構成要素を包含するＲＳＡ署名装置（べき乗剰余計算装置）３３０を備えている。なお、図４の構成例の動作は、基本的には、図１の構成例と同様である。

【００８５】
前者の図１の構成例の場合には、例えば、鍵発行側が外部鍵発行機１１０及び分割鍵生成装置１２０を備え、各利用者が例えばＩＣカードなどからなるＲＳＡ署名装置１３０を利用するような形態が考えられる。この場合には、例えば、まず、鍵発行側において外部鍵発行機１１０及び分割鍵生成装置１２０により、当該利用者に応じた各種鍵をＲＳＡ署名装置１３０に書き込み、該利用者は鍵発行側から該各種鍵を書き込まれたＲＳＡ署名装置１３０を取得し、これを計算機等の装置に装着するなどして使用する。

【００８６】
この場合、ＲＳＡ署名装置１３０に書き込まれた鍵は、固定、または半固定で使用することが想定される。半固定にする場合、鍵の変更は、鍵発行側に行って貰う。半固定にする場合の鍵変更時の処理（データのやり取り）の手段としては、例えば、ＲＳＡ署名装置１３０を現実に鍵発行側へ渡して分割鍵生成装置１２０により鍵の変更（新たな分割秘密鍵の書き込み等）を行って貰う方法や、ＲＳＡ署名装置１３０と分割鍵生成装置１２０との間をネットワークを介して接続可能にして鍵の変更（新たな分割秘密鍵の書き込み等）を行って貰う方法などがある。ネットワークを介する場合、ＲＳＡ署名装置１３０と分割鍵生成装置１２０との間でのやり取りには、認証や暗号化などの技術を利用するのが望まし

い。
【００８７】
なお、この利用形態の場合には、外部鍵発行機１１０及び分割鍵生成装置１２０を一体化して構成してもよい。
【００８８】
後者の図４の構成例の場合には、例えば、鍵発行側が外部鍵発行機２１０を備え、各利用者が例えばＩＣカードなどからなるＲＳＡ署名装置２２０を利用するような形態が考えられる。この場合には、例えば、鍵発行側において外部鍵発行機２１０により、当該利用者に応じた秘密鍵及び公開鍵をＲＳＡ署名装置２２０に書き込み、該利用者は鍵発行側から該秘密鍵及び公開鍵を書き込まれたＲＳＡ署名装置２２０を取得し、該ＲＳＡ署名装置２２０これを計算機等の装置に装着するなどして使用する。なお、図１の構成例で、例えば、鍵発行側が外部鍵発行機１１０を備え、各利用者が例えばＩＣカードなどからなる分割鍵生成装置１２０及びＲＳＡ署名装置１３０を利用するような場合も同様である。
【００８９】
この場合には、鍵インデックス値、分割秘密鍵とダミー鍵の値、鍵の分割方法などを、例えば署名生成の処理毎など、上記の半固定の利用形態よりもきめ細かい単位毎に（タイミングで）、変更することができ、更なるセキュリティ強度の向上が期待される。
【００９０】
なお、図１や図４とは異なる構成例も可能である。例えば、図１において、乱数生成部１２２をＲＳＡ署名装置１３０側に備える構成も可能である。
【００９１】
ところで、これまで図１や図４を参照しながら説明してきた構成例において、秘密鍵の分割方法や分割秘密鍵の使用順序などの分割鍵に関係する要素の少なくとも一部を可変として、ダミー鍵を使わずに、分割秘密鍵だけ用いて処理をするようにしてもよい。これによっても、秘密鍵の分割方法あるいは分割秘密鍵の使用順序などが不明であるという点で、セキュリティ強度を上げることができる。
【００９２】
ダミー鍵を使用しない場合には、図２の分割秘密鍵生成手順については、ステップＳ２，Ｓ３を省き、ステップＳ４でｔ［０］を求める部分を省き、ステップＳ５でダミー鍵ｄｕ［ｕ］と鍵インデックスｔ［ｉ］を出力する部分を省く修正をすればよい。また、図３のＲＳＡ署名手順は、図５のように修正すればよい。
【００９３】
なお、ダミー鍵を使用する使用しないかを、所定のタイミングで（例えば、毎回）、選択できる構成にしてもよい。
【００９４】
また、常にダミー鍵を使用しない構成にする場合には、図１や図４の乱数生成部については、ダミー鍵および鍵インデックスを生成する機能が不要になり、ダミーの計算結果Ｓ’を保持するデータ記憶部（０）は、それ自体が不要になる。
【００９５】
（第２の実施形態）
次に、本発明の第２の実施形態として、中国人剰余定理を使用する場合の実施形態について説明する。
【００９６】
第１の実施形態では、Ｍ＾ｄ　ｍｏｄ　ｎの計算において、ｄに分割秘密鍵の手法を適用したが、本実施形態では、Ｍ＾ｄ　ｍｏｄ　ｎの計算に中国人剰余定理を使用する場合のＳｐ＝Ｍｐ＾ｄｐ　ｍｏｄ　ｐおよびＳｑ＝Ｍｑ＾ｄｑ　ｍｏｄ　ｑにおけるｄｐとｄｑの個々に分割秘密鍵の手法を適用するものである。したがって、本実施形態は、第１の実施形態を、中国人剰余定理を使用するように修正したものであり、分割秘密鍵の手法に関係する部分は、基本的には、第１の実施形態と同様である（第１の実施形態の変形例の説明も、本実施形態について同様に当て嵌まる）。

【００９７】
図６に、本実施形態に係る暗号処理システムの構成例を示す。図６は、分割鍵生成装置とＲＳＡ署名装置（べき乗剰余計算装置）とが分離している分離型の構成をとる場合の一例である。
【００９８】
図６に示されるように、本暗号処理システムは、外部鍵発行機３１０、分割鍵生成装置３２０、ＲＳＡ署名装置３３０を備えている。また、分割鍵生成装置３２０は、ＲＳＡ鍵記憶部３２１、乱数生成部３２２、分割鍵生成部３２３を含む。ＲＳＡ署名装置３３０は、分割鍵記憶部３３１、べき乗剰余計算部３３２、第１のデータ記憶部（以下、データ記憶部（０）と記述する）３３３、第２のデータ記憶部（以下、データ記憶部（１）と記述する）３３４、平文分割部３３５、ＣＲＴ合成部３３６、出力部３３７を含む。
【００９９】
外部鍵発行機３１０は、ＲＳＡの公開鍵（ｅ，ｎ）、秘密鍵（ｄ，ｄｐ，ｄｑ，ｐ，ｑ）を生成し、分割鍵生成装置３２０内のＲＳＡ鍵記憶部３２１に供給する。ただし、ｐ，ｑはそれぞれ大きな素数で、ｎ＝ｐ＊ｑ，　ｅｄ＝＝１　（ｍｏｄ　ｌｃｍ（ｐ－１）（ｑ－１））を満たし、ｄｐ＝ｄ　ｍｏｄ　（ｐ－１），ｄｑ＝ｄ　ｍｏｄ　（ｑ－１）である。
【０１００】
分割鍵生成装置３２０は、ｄｐに対する分割秘密鍵及びｄｑに対する分割秘密鍵等を生成してＲＳＡ署名装置３３０に供給し、ＲＳＡ署名装置３３０は、供給されたそれら分割秘密鍵等を用いて中国人剰余定理を使用して平文Ｍに対する署名文Ｍ＾ｄ　ｍｏｄ　ｎを求めるものである。
【０１０１】
まず、分割鍵生成装置３２０について説明する。
【０１０２】
ＲＳＡ鍵記憶部３２１は、外部鍵発行機３１０から供給されるＲＳＡの公開鍵（ｅ，ｎ）と秘密鍵（ｄ，ｄｐ，ｄｑ，ｐ，ｑ）を保持し、必要に応じて分割鍵生成部３２４に公開鍵（ｅ，ｎ）と秘密鍵（ｄ，ｄｐ，ｄｑ，ｐ，ｑ）を供給する。
【０１０３】
乱数生成部３２２は、任意のビット長の鍵およびビット列を任意の数だけ乱数により生成し、それらを分割鍵生成部３２３に供給する。
【０１０４】
分割鍵生成部３２３は、乱数生成部３２２から供給される法ｐ上の分割秘密鍵の構成成分と法ｑ上の分割秘密鍵の構成成分と、ＲＳＡ鍵記憶部３２１から供給される秘密鍵（ｄ，ｄｐ，ｄｑ，ｐ，ｑ）、公開鍵（ｅ，ｎ）を用いて法ｐ上の分割秘密鍵と法ｑ上の分割秘密鍵を生成する。
【０１０５】
以下、中国人剰余定理を使用する場合の分割秘密鍵生成手順について説明する。
【０１０６】
図７に、この場合の処理手順の一例を示す。
【０１０７】
ここでは、分割秘密鍵生成の一例として、秘密鍵ｄｐを乗法のみによりk_p個に分割し、秘密鍵ｄｑを乗法のみによりk_q個に分割するものとする。すなわち、
$dp = dp[0] * dp[1] * \cdots * dp[k_p - 1] \mod (p-1)$，
$dq = dq[0] * dq[1] * \cdots * dq[k_q - 1] \mod (q-1)$
となるように、ｄｐをｄｐ［０］，…，ｄｐ［k_p－１］に分割し、ｄｑをｄｑ［０］，…，ｄｑ［k_q－１］に分割する。なお、以下では、$k_p = k_q = k$として説明する。
【０１０８】
まず、乱数生成部３２２は、ｓビットの乱数を$k_p - 1 = k - 1$個生成し、それぞれ、分割秘密鍵ｄｐ［１］～ｄｐ［ｋ－１］へ代入する（ステップＳ２１）。なお、ｓは、秘密

鍵ｄのビット長より小さいビット数（秘密鍵ｄが例えば５１２ビットあるいは１０２４ビット程度の場合に分割秘密鍵ｄｐ［１］～ｄｐ［ｋ－１］がそれぞれ例えば８ビット程度）にすると好ましい。

【０１０９】

また、乱数生成部３２２は、ｓビットの乱数を$k_q-1=k-1$個生成し、それぞれ、分割秘密鍵ｄｑ［１］～ｄｑ［ｋ－１］へ代入する（ステップＳ２２）。

【０１１０】

また、乱数生成部３２２は、$(k_p-1)+(k_q-1)=2(k-1)$ビット長で且つハミング重みを$k_p-1=k-1$とする鍵インデックスｔ［ｉ］　（２≦ｉ≦２ｋ－１）をランダムに生成する（ステップＳ２３）。

【０１１１】

次いで、分割鍵生成部３２３は、ｔ［ｉ］＝１となる鍵（つまり、分割秘密鍵ｄｐ［ｊ］（１≦ｊ≦ｋ－１））を掛け合わせて、

Ｄｐ＝ｄｐ［１］＊ｄｐ［２］＊…＊ｄｐ［ｋ－１］

とし、ＲＳＡ鍵記憶部３２１から供給される秘密鍵ｄｐを用いて、

ｄｐ［０］＝ｄｐ＊Ｄｐ^－１　ｍｏｄ　（ｐ－１）

とすることで、法ｐ上の残り１つの分割秘密鍵ｄｐ［０］を求める（ステップＳ２４）。

【０１１２】

同様にして、分割鍵生成部３２３は、ｔ［ｉ］＝０となる鍵（つまり、分割秘密鍵ｄｑ［ｊ］（１≦ｊ≦ｋ－１））を掛け合わせて、

Ｄｑ＝ｄｑ［１］＊ｄｑ［２］＊…＊ｄｑ［ｋ－１］

とし、ＲＳＡ鍵記憶部３２１から供給される秘密鍵ｄｑを用いて、

ｄｑ［０］＝ｄｑ＊Ｄｑ^－１　ｍｏｄ　（ｑ－１）

とすることで、法ｑ上の残り１つの分割秘密鍵ｄｑ［０］を求める（ステップＳ２４）。また、ｔ［０］＝１、ｔ［１］＝０とする（ステップＳ２４）。なお、その逆に、ｔ［０］＝０、ｔ［１］＝１としてもよい。

【０１１３】

以上で、

法ｐ上の分割秘密鍵ｄｐ［ｊ］（０≦ｊ≦ｋ－１）、

法ｑ上の分割秘密鍵ｄｑ［ｕ］（０≦ｕ≦ｋ－１）、

鍵インデックスｔ［ｉ］（０≦ｉ≦２ｋ－１）

が完成する。

【０１１４】

そして、分割鍵生成部３２３は、

ｄｐ［ｊ］（０≦ｊ≦ｋ－１）、

ｄｑ［ｕ］（０≦ｕ≦ｋ－１）、

ｔ［ｉ］（０≦ｉ≦２ｋ－１）、

公開鍵（ｅ，ｎ）、

秘密鍵（ｄ，ｄｐ，ｄｑ，ｐ，ｑ）

をＲＳＡ署名装置３３０へ出力する（ステップＳ２５）。

【０１１５】

なお、図７の手順においては、ｄｐ［１］，ｄｐ［２］，…，ｄｐ［ｋ－１］を全て生成した後にｄｐ［０］を求める必要があり、ｄｑ［１］，ｄｑ［２］，…，ｄｑ［ｋ－１］を全て生成した後にｄｑ［０］を求める必要があるが、それ以外は、法ｐ上のｋ個の分割秘密鍵と法ｑ上のｋ個の分割秘密鍵と鍵インデックスをどのような順番で求めても構わない。

【０１１６】

次に、ＲＳＡ署名装置３３０について説明する。

【０１１７】

分割鍵記憶部３３１は、分割鍵生成装置３２０内の分割鍵生成部３２３から供給される法

ｐ上の分割秘密鍵、法ｑ上の分割秘密鍵、鍵インデックス、公開鍵（ｅ，ｎ）、秘密鍵（ｄ，ｄｐ，ｄｑ，ｐ，ｑ）を保持し、必要に応じてそれぞれをべき乗剰余計算部３３２に供給する。

【０１１８】

データ記憶部（０）３３３には、法ｑ上のｋ個の分割秘密鍵を使用したべき乗剰余計算の計算結果Ｓｑが格納され、データ記憶部（１）３３４には、法ｐ上のｋ個の分割秘密鍵を使用したべき乗剰余計算の計算結果Ｓｐが格納される。なお、Ｓｐは、最終的に、Ｍｐ＾ｄｐ　ｍｏｄ　ｐを与え、Ｓｑは、最終的に、Ｍｑ＾ｄｑ　ｍｏｄ　ｑを与えることになるものである。ここで、Ｍｐ＝Ｍ　ｍｏｄｐ、Ｍｑ＝Ｍ　ｍｏｄ　ｑである。

【０１１９】

平文分割部３３５は、対象となる平文Ｍをもとに、Ｍｐ＝Ｍ　ｍｏｄ　ｐおよびＭｑ＝Ｍ　ｍｏｄ　ｑを生成し、データ記憶部（０）３３３の初期値としてＭｑを供給するとともに、データ記憶部（１）３３４の初期値としてＭｐを供給する。

【０１２０】

べき乗剰余計算部３３２は、分割鍵記憶部３３１から供給される鍵インデックス値ｔ［ｉ］が１ならば、データ記憶部（１）３３４から前回までの計算結果Ｓｐ（ただし、初回は初期値Ｓｐ＝Ｍ　ｍｏｄ　ｐとなる）を呼び出して、法ｐ上の分割秘密鍵ｄｐ［ｊ］を使ってＳｐ＝Ｓｐ＾ｄｐ［ｊ］　ｍｏｄ　ｐの計算を行い、その結果Ｓｐをデータ記憶部（１）３３４に格納する。他方、鍵インデックス値ｔ［ｉ］が０ならば、データ記憶部（０）３３３から前回までの計算結果Ｓｑ（ただし、初回は初期値Ｓｑ＝Ｍ　ｍｏｄ　ｑとなる）を呼び出して、法ｑ上の分割秘密鍵ｄｑ［ｕ］を使ってＳｑ＝Ｓｑ＾ｄｑ［ｕ］　ｍｏｄ　ｑの計算を行い、その結果Ｓｑをデータ記憶部（０）３３３に格納する。

【０１２１】

なお、ここでは、鍵インデックス値ｔ［ｉ］＝１の場合に、法ｐ上の分割秘密鍵によるべき乗剰余計算を行い、鍵インデックス値ｔ［ｉ］＝０の場合に、法ｑ上の分割秘密鍵によるべき乗剰余計算を行うようにしているが、これに限定されるものではなく、例えば、鍵インデックス値ｔ［ｉ］＝０の場合に、法ｐ上の分割秘密鍵によるべき乗剰余計算を行い、鍵インデックス値ｔ［ｉ］＝１の場合に、法ｑ上の分割秘密鍵によるべき乗剰余計算を行うようにしてもよい。

【０１２２】

ＣＲＴ合成部３３６は、データ記憶部（０）３３３の最終結果Ｓｑとデータ記憶部（１）３３４の最終結果ＳｐとをＳ＝｛Ｓｐ＊ｑ＊（ｑ＾－１　ｍｏｄ　ｐ）＋Ｓｑ＊ｐ＊（ｐ＾－１　ｍｏｄ　ｑ）｝　ｍｏｄ　ｎなる処理で合成し、合成結果Ｓ（＝Ｍ＾ｄ　ｍｏｄ　ｎ）を出力部３３７に供給する。

【０１２３】

出力部３３７は、最終署名結果としてＳ（＝Ｍ＾ｄ　ｍｏｄ　ｎ）を出力する。

【０１２４】

以下、中国人剰余定理を使用する場合の分割秘密鍵を用いたＲＳＡ署名手順について説明する。

【０１２５】

図８に、この場合の処理手順の一例を示す。

【０１２６】

なお、ここでは、法ｐ乗の分割秘密鍵ｄｐ［ｊ］は、ｊの値の小さい方から逐次使用し、法ｑ乗の分割秘密鍵ｄｑ［ｕ］は、ｕの値の小さい方から逐次使用するものとした例を示している。

【０１２７】

最初に、データ記憶部（０）３３３およびデータ記憶部（１）３３４にそれぞれ初期値として平文分割部３３５から供給されるＭｑ及びＭｐを書き込み、また、ｉ＝０、ｊ＝０、ｕ＝０とする（ステップＳ３１）。

【０１２８】

以降は、$k_p + k_q = 2k$回の繰り返し処理となる。

【０１２９】

まず、べき乗剰余計算部３３２は、鍵インデックス値ｔ［ｉ］を参照し、ｔ［ｉ］＝１ならば（ステップＳ３２）、データ記憶部（１）３３４からＳｐを読み出し、法ｐ上の分割秘密鍵ｄｐ［ｊ］を使って、Ｓｐ＝Ｓｐ＾ｄｐ［ｊ］　ｍｏｄ　ｐの計算を行い、その結果Ｓｐをデータ記憶部（１）３３４に格納し（ステップＳ３３）、ｊおよびｉをそれぞれ１ずつインクリメントする（ステップＳ３４，Ｓ３７）。

【０１３０】

他方、鍵インデックス値ｔ［ｉ］が０ならば（ステップＳ３２）、データ記憶部（０）３３３からＳｑを読み出し、法ｑ上の分割秘密鍵ｄｑ［ｕ］を使って、Ｓｑ＝Ｓｑ＾ｄｑ［ｕ］　ｍｏｄ　ｑの計算を行い、その結果Ｓｑをデータ記憶部（０）３３３に格納し（ステップＳ３５）、ｕおよびｉをそれぞれ１ずつインクリメントする（ステップＳ３６，Ｓ３７）。

【０１３１】

上記の２系統のうちの鍵インデックス値ｔ［ｉ］に応じたいずれかが行われた後に、ｉの値を参照し、ｉの値が法ｐ上の分割秘密鍵と法ｑ上の分割秘密鍵の総数２ｋを超えていないなら（ステップＳ３８）、ステップＳ３２に戻り、次のループの処理を行う。

【０１３２】

最終的に、ステップＳ３８において、ｉ＞２ｋとなったならば、処理ループを抜けて、ＣＲＴ合成部３３６により、データ記憶部（０）３３３の最終結果Ｓｑとデータ記憶部（１）３３４の最終結果ＳｐとをＳ＝｛Ｓｐ＊ｑ＊（ｑ＾－１　ｍｏｄ　ｐ）＋Ｓｑ＊ｐ＊（ｐ＾－１　ｍｏｄ　ｑ）｝　ｍｏｄ　ｎなる処理で合成して、平文Ｍに対する署名文Ｓ＝Ｍ＾ｄ　ｍｏｄ　ｎを求める（ステップＳ３９）。

【０１３３】

以上のように処理を行うことで、中国人剰余定理を使用する場合にも、法ｐ上の分割秘密鍵によるべき乗剰余計算と法ｑ上の分割秘密鍵によるべき乗剰余計算とをランダムな順序に実行させて、電力解析攻撃を困難にさせる。

【０１３４】

このように本実施形態においては、電力解析攻撃によってＲＳＡ署名処理中の消費電力波形から、たとえ分割秘密鍵の全ての値が分かったとしても、法ｐ上の分割秘密鍵と法ｑ上の分割秘密鍵との区別が特定されなければ、秘密鍵ｄｐおよびｄｑを特定することができない。したがって、秘密鍵を求めるために必要な総当り数（測定される波形の個数ｋｐ＋ｋｑから法ｐ上の分割秘密鍵の個数ｋｐを選び出す組合せ）を大きくすることで、計算量的に攻撃を困難にすることができる。これまで説明した構成例では、鍵の分割数ｋｐやｋｑを多くすればするほど、セキュリティ強度が上がる。

【０１３５】

ちなみに、中国人剰余定理を使用して、秘密鍵ｄｐ，ｄｑを６４個（上記の例でいうとｋｐ＝ｋｑ＝ｋ＝３２）に分割した場合は、約２の６０乗の鍵の組み合わせを考慮しなければ、秘密鍵は求まらないことになる。また、これに加えて、後述するように、例えば鍵の分割方法に関する情報なども可変とすると、その情報も必要となってくるため、秘密鍵の情報を得ることは非常に困難となる。

【０１３６】

以下では、本実施形態の変形例について説明する。

【０１３７】

上記では、法ｐ上の分割秘密鍵ｄｐ［ｊ］は、ｊの値の小さい方から逐次使用し、法ｑ上の分割秘密鍵ｄｑ［ｕ］は、ｕの値の小さい方から逐次使用するものとしたが、もちろん、これらとは異なる順番で鍵を使用していってもよい。

【０１３８】

また、上記では、法ｐ上の分割秘密鍵ｄｐ［ｊ］の使用順序および法ｑ上の分割秘密鍵ｄｑ［ｕ］の使用順序は固定のものであったが、法ｐ上の分割秘密鍵の使用順序と法ｑ上の

分割秘密鍵の使用順序の一方又は双方を、所定のタイミングで（例えば、毎回）、ランダムに変えるようにしてもよい。このように鍵の使用順序をランダムにすることによって、さらにセキュリティ強度を上げることができる。

【０１３９】

また、上記では、秘密鍵ｄｐやｄｑの分割数ｋｐやｋｑは固定であったが、分割数ｋｐとｋｑの一方又は双方を可変としてもよい。このように分割数を可変とすることによって、さらにセキュリティ強度を上げることができる。

【０１４０】

また、上記では、秘密鍵ｄｐやｄｑの分割方法として、乗法のみのものを使用したが、他の分割方法、例えば、加法のみの分割方法、減法のみの分割方法、除法のみの分割方法も可能であり、また、加法・減法・乗法・除法の適当な組合せも可能である。

【０１４１】

また、上記では、秘密鍵ｄｐやｄｑの分割方法は特定のものに固定されていたが、例えば、装置（あるいはユーザ）毎に秘密鍵の分割方法を変える（同一の装置（あるいはユーザ）では秘密鍵の分割方法は固定する）ようにしてもよいし、また、例えば、同一の装置（あるいはユーザ）についても、所定のタイミングで（例えば、毎回）、鍵の分割方法を変えるようにしてもよい。このように秘密鍵の分割方法をランダムにすることによって、さらにセキュリティ強度を上げることができる。

【０１４２】

また、秘密鍵ｄｐと秘密鍵ｄｑとで、秘密鍵の分割方法を同一にしてもよいし、異なる秘密鍵の分割方法を用いるようにしてもよい。

【０１４３】

また、図６は、分割鍵生成装置とＲＳＡ署名装置とが分離している分離型の構成をとる場合の一例であったが、分割鍵生成装置とＲＳＡ署名装置が一体化している一体型の構成をとることも可能である。図９に、この場合の暗号処理システムの構成例を示す。図９に示されるように、本暗号処理システムは、図６の外部鍵発行機３１０と同様の外部鍵発行機４１０と、図６の分割鍵生成装置３２０の構成要素とＲＳＡ署名装置３３０の構成要素を包含するＲＳＡ署名装置（べき乗剰余計算装置）４２０を備えている。なお、図９の構成例の動作は、基本的には、図６の構成例と同様である。図６や図９の構成例に関する利用形態の例については、第１の実施形態で図１および図４を参照しながら説明したものと同様である。

【０１４４】

また、なお、図６や図９とは異なる構成例も可能である。例えば、図６において、乱数生成部３２２をＲＳＡ署名装置３３０側に備える構成も可能である。

【０１４５】

ところで、これまで図６や図９を参照しながら説明してきた構成例において、さらに第１の実施形態のように、ダミー鍵を法ｐ上の分割秘密鍵、法ｑ上の分割秘密鍵に混ぜ込んで処理をするようにしてもよい。

【０１４６】

ダミー鍵を使用しない場合には、図６のＲＳＡ署名装置３３０や図９のＲＳＡ署名装置４２０において、データ記憶部（０）およびデータ記憶部（１）に加えて、ダミー鍵を使用したべき乗剰余計算の計算結果（すなわちダミーの計算結果）Ｓ’を格納するための第３のデータ記憶部（以下、データ記憶部（２）と記述する）をさらに設ける。図６のＲＳＡ署名装置３３０においてさらにデータ記憶部（２）３３８を設けた様子を図１０に示す。

【０１４７】

なお、鍵インデックスｔ［ｉ］の代わりに、２ビット単位の鍵インデックスｔｔ［ｉ］を考え、例えば、ｔｔ［ｉ］＝“０１”の場合には、法ｐ上の分割秘密鍵によるべき乗剰余計算の処理を行い、ｔｔ［ｉ］＝“００”の場合には、法ｑ上の分割秘密鍵によるべき乗剰余計算の処理を行い、ｔｔ［ｉ］＝“１０”ｏｒ“１１”の場合には、ダミー鍵によるべき乗剰余計算の処理を行うようにする。なお、鍵インデックスｔ［ｉ］がどのような値

のときに、３つのうちのいずれのべき乗剰余計算の処理を行うかについての対応は、上記の例に限定されるものではなく、他の対応でも構わない。

【０１４８】

また、ダミー鍵を使用する場合には、図７の分割秘密鍵生成手順については、ステップＳ２５以前の適当なタイミングにｘ個のダミー鍵ｄｕ［ｕ］（０≦ｕ≦ｘ－１）を生成するステップを設ける。また、ステップＳ２３では、（ｋｐ＋ｋｑ＋ｘ－２）＊２ビット長で且つ上記の例の場合には“０１”なる２ビットをｋｐ－１個含み且つ上記の例の場合には“００”なる２ビットをｋｑ－１個含む鍵インデックスｔｔ［ｉ］（２≦ｉ≦ｋｐ＋ｋｑ＋ｘ－１）をランダムに生成する。また、ステップＳ２４では、ｔｔ［０］＝“０１”、ｔｔ［１］＝“００”とするか、あるいは、その逆に、ｔｔ［０］＝“００”、ｔｔ［１］＝“０１”とする。

【０１４９】

また、図８のＲＳＡ署名手順は、図１１のように修正すればよい。図１１は、図８の手順と比較して、鍵インデックスｔ［ｉ］の値（２ビット）で、３系統に分岐する。すなわち、ダミー鍵によるべき乗剰余計算の系統が加えられている。なお、ステップＳ４０では、ｔｔ［ｉ］＝“１０”のときにＳ’＝Ｓｐ＾ｄｕ［ｕ］ ｍｏｄ ｐを行い、ｔｔ［ｉ］＝“１１”のときにＳ’＝Ｓｑ＾ｄｕ［ｕ］ ｍｏｄ ｑを行うようにしているが、ｔｔ［ｉ］の値にかかわらずに、Ｓ’＝Ｓｐ＾ｄｕ［ｕ］ ｍｏｄ ｐまたはＳ’＝Ｓｑ＾ｄｕ［ｕ］ ｍｏｄ ｑの一方のみを行うようにしてもよいし、Ｓ’＝Ｓｐ＾ｄｕ［ｕ］ ｍｏｄ ｐまたはＳ’＝Ｓｑ＾ｄｕ［ｕ］ ｍｏｄ ｑ以外の処理を行う構成も可能である。

【０１５０】

なお、ダミー鍵を使用するか使用しないかを、所定のタイミングで（例えば、毎回）、選択できる構成にしてもよい。

【０１５１】

また、常にダミー鍵を使用しない構成にする場合には、図６や図９の乱数生成部については、ダミー鍵および鍵インデックスを生成する機能が不要になり、ダミーの計算結果Ｓ’を保持するデータ記憶部（２）は、それ自体が不要になる。

【０１５２】

なお、第１の実施形態の構成と第２の実施形態の構成とを両方備え、中国人剰余定理を使用するか使用しないかを選択可能にしてもよい。

【０１５３】

さて、これまでは、ＲＳＡ暗号の署名生成の場合を例にとって説明したが、もちろん、ＲＳＡの復号の場合についても同様に可能である。この場合には、これまで説明してきたＲＳＡ署名装置がそのままＲＳＡ復号装置となる。その際、入力を暗号文Ｍ＾ｅ ｍｏｄｎとすればよく、出力としては平文Ｍ（＝（Ｍ＾ｅ ｍｏｄ ｎ）＾ｄ ｍｏｄ ｎ）が得られる。

【０１５４】

また、本実施形態では、公開鍵暗号方式の一例としてＲＳＡ暗号の場合を例にとって説明したが、本発明は、その他の公開鍵暗号方式（楕円曲線暗号、ＤＨ鍵共有、ＥｌＧａｍａｌ暗号等）にも適用可能である。

【０１５５】

なお、本実施形態の各々の装置は、ハードウェアとしてもソフトウェアとして実現可能である。

【０１５６】

また、本実施形態の各々の装置は、コンピュータに所定の手段を実行させるための（あるいはコンピュータを所定の手段として機能させるための、あるいはコンピュータに所定の機能を実現させるための）プログラムとして実施することもでき、該プログラムを記録したコンピュータ読取り可能な記録媒体として実施することもできる。

【０１５７】

また、本実施形態の各々の装置は、独立した装置としても、例えば計算機等への組み込み用の装置としても、実施可能である。また、例えば、本実施形態の装置をＩＣカードで実現した場合に、該ＩＣカードをＣＰＵ及びメモリ内蔵のものとし、ＩＣカードに組み込むソフトウェアとして実現することも可能である。

【０１５８】

その他にも種々の実現形態が可能である。

【０１５９】

なお、この発明の実施の形態で例示した構成は一例であって、それ以外の構成を排除する趣旨のものではなく、例示した構成の一部を他のもので置き換えたり、例示した構成の一部を省いたり、例示した構成に別の機能あるいは要素を付加したり、それらを組み合わせたりすることなどによって得られる別の構成も可能である。また、例示した構成と論理的に等価な別の構成、例示した構成と論理的に等価な部分を含む別の構成、例示した構成の要部と論理的に等価な別の構成なども可能である。また、例示した構成と同一もしくは類似の目的を達成する別の構成、例示した構成と同一もしくは類似の効果を奏する別の構成なども可能である。

また、この発明の実施の形態で例示した各種構成部分についての各種バリエーションは、適宜組み合わせて実施することが可能である。

また、この発明の実施の形態は、個別装置としての発明、関連を持つ２以上の装置についての発明、システム全体としての発明、個別装置内部の構成部分についての発明、またはそれらに対応する方法の発明等、種々の観点、段階、概念またはカテゴリに係る発明を包含・内在するものである。

従って、この発明の実施の形態に開示した内容からは、例示した構成に限定されることなく発明を抽出することができるものである。

【０１６０】

本発明は、上述した実施の形態に限定されるものではなく、その技術的範囲において種々変形して実施することができる。

【０１６１】

【発明の効果】

本発明によれば、公開鍵暗号方式に対する電力解析攻撃による秘密鍵漏洩を防ぐことができる。

【図面の簡単な説明】

【図１】本発明の第１の実施形態に係る暗号処理システムの構成例を示す図

【図２】同実施形態に係る分割秘密鍵生成手順の一例を示すフローチャート

【図３】同実施形態に係る分割秘密鍵を用いたＲＳＡ署名手順の一例を示すフローチャート

【図４】同実施形態に係る暗号処理システムの他の構成例を示す図

【図５】同実施形態に係る分割秘密鍵を用いたＲＳＡ署名手順の他の例を示すフローチャート

【図６】本発明の第２の実施形態に係る暗号処理システムの構成例を示す図

【図７】同実施形態に係る分割秘密鍵生成手順の一例を示すフローチャート

【図８】同実施形態に係る分割秘密鍵を用いたＲＳＡ署名手順の一例を示すフローチャート

【図９】同実施形態に係る暗号処理システムの他の構成例を示す図

【図１０】同実施形態に係る暗号処理システムのさらに他の構成例を示す図

【図１１】同実施形態に係る分割秘密鍵を用いたＲＳＡ署名手順の他の例を示すフローチャート

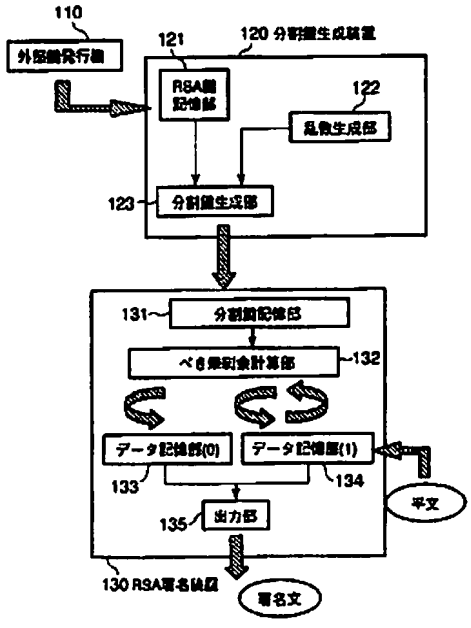
【図１２】ｓｑｕａｒｅ－ｍｕｌｔｉｐｌｙ　ｍｅｔｈｏｄの処理手順を示すフローチャート

【符号の説明】

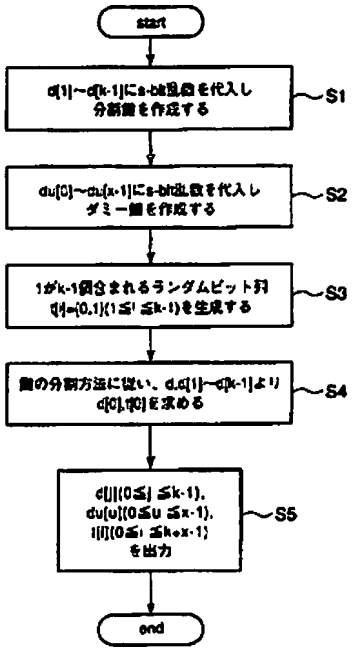
１１０，２１０，３１０，４１０…外部鍵発行機

１２０，３２０…分割鍵生成装置
１３０，２２０，３３０，４２０…ＲＳＡ署名装置
１２１，２２１，３２１，４２１…ＲＳＡ鍵記憶部
１２２，２２２，３２２，４２２…乱数生成部
１２３，２２３，３２３，４２３…分割鍵生成部
１３１，２２４，３３１，４２４…分割鍵記憶部
１３２，２２５，３３２，４２５…べき乗剰余計算部
１３３，１３４，２２６，２２７，３３３，３３４，３３８，４２６，４２７…データ記憶部
１３５，２２８，３３７，４３０…出力部
３３５，４２８…平文分割部
３３６，４２９…ＣＲＴ合成部

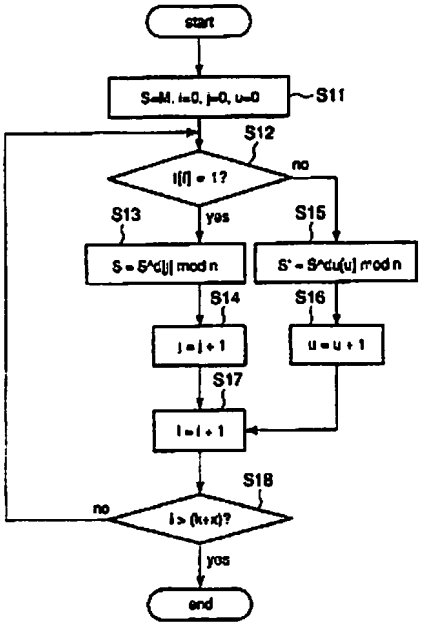
【図１】



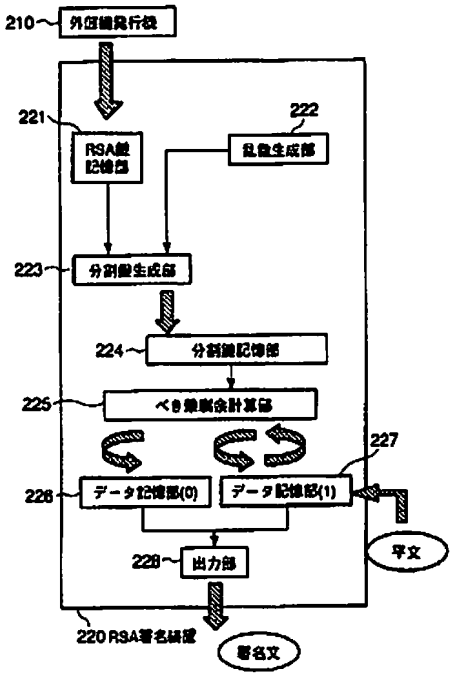
【図２】



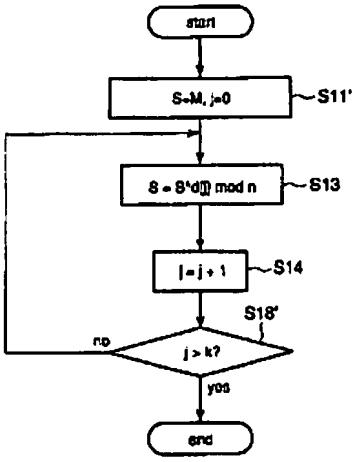
【図３】


start
S=M, i=0, j=0, u=0
S11
S12
I[i] = 1?
no
yes
S13
S15
S = S^d[j] mod n
S' = S^du[u] mod n
S14
S16
j = j + 1
u = u + 1
S17
i = i + 1
S18
no
i > (k+x)?
yes
end

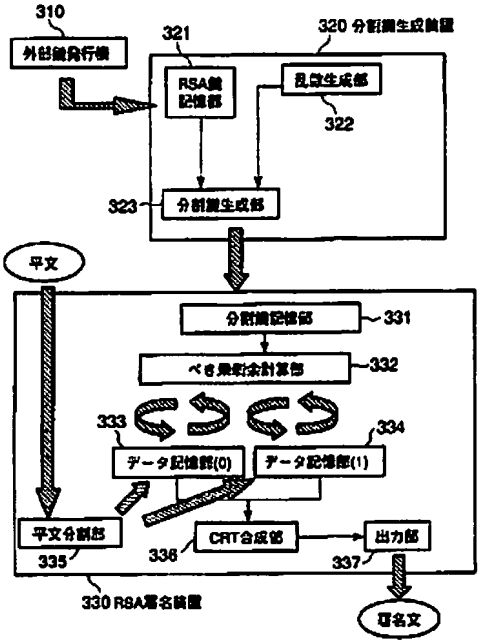
【図４】


210
外部鍵発行機
221
RSA鍵記憶部
222
乱数生成部
223
分割鍵生成部
224
分割鍵記憶部
225
べき乗剰余計算部
227
226
データ記憶部(0)
データ記憶部(1)
平文
228
出力部
220 RSA署名装置
署名文

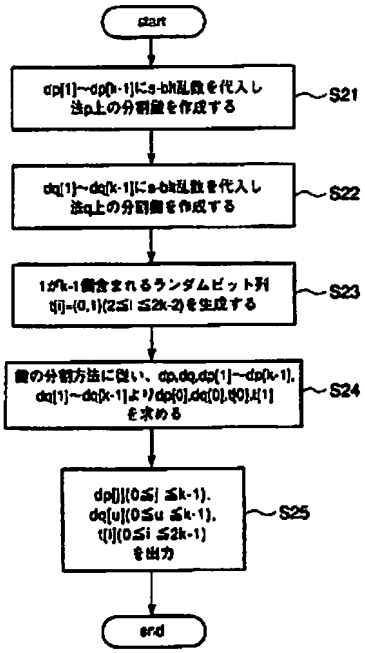
【図５】


start
S=M, j=0
S11'
S = S^d[j] mod n
S13
j = j + 1
S14
S18'
no
j > k?
yes
end

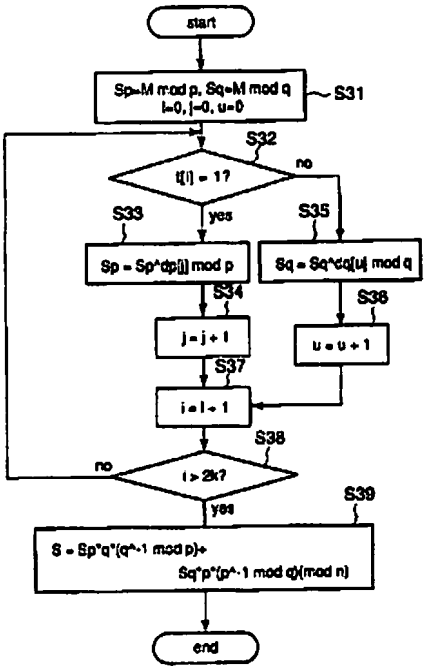
【図６】


310
外部鍵発行機
321
320 分割鍵生成装置
RSA鍵記憶部
乱数生成部
322
323
分割鍵生成部
平文
分割鍵記憶部
331
べき乗剰余計算部
332
333
334
データ記憶部(0)
データ記憶部(1)
平文分割部
CRT合成部
出力部
335
336
337
330 RSA署名装置
署名文

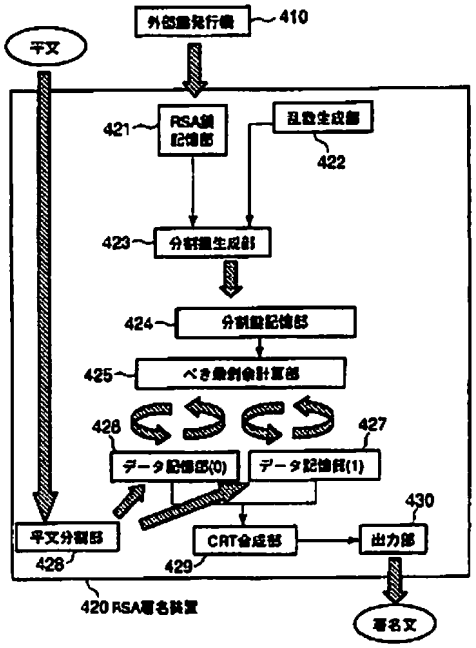
【図７】


start
dp[1]～dp[k-1]にs-bit乱数を代入し
法p上の分割鍵を作成する
S21
dq[1]～dq[k-1]にs-bit乱数を代入し
法q上の分割鍵を作成する
S22
1がk-1個含まれるランダムビット列
t[i]=(0,1)(2≦i ≦2k-2)を生成する
S23
前の分割方法に従い、dp,dq,dp[1]～dp[k-1],
dq[1]～dq[k-1]よりdp[0],dq[0],t[0],t[1]
を求める
S24
dp[j](0≦j ≦k-1),
dq[u](0≦u ≦k-1),
t[i](0≦i ≦2k-1)
を出力
S25
end

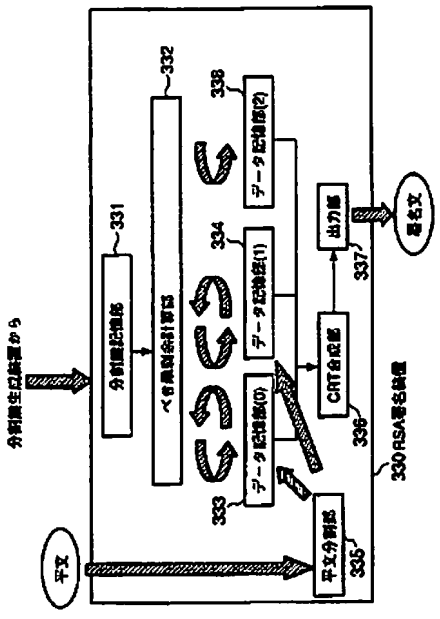
【図８】


start
Sp=M mod p, Sq=M mod q
i=0, j=0, u=0
S31
S32
t[i] = 1?
no
yes
S33
S35
Sp = Sp*dp[j] mod p
Sq = Sq*dq[u] mod q
S34
S36
j = j + 1
u = u + 1
S37
i = i + 1
S38
no
i > 2k?
yes
S39
S = Sp*q*(q^-1 mod p)+
Sq*p*(p^-1 mod q)(mod n)
end

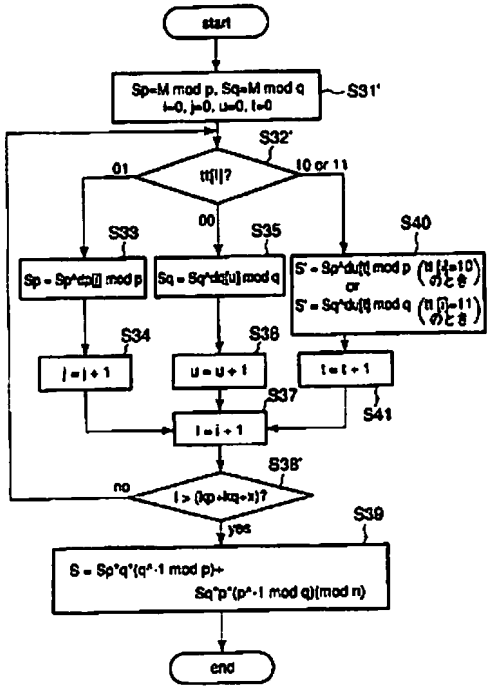
【図９】


平文
外部鍵発行機
410
RSA鍵
記憶部
421
乱数生成部
422
分割鍵生成部
423
分割鍵記憶部
424
べき剰余計算部
425
428
427
データ記憶部(0)
データ記憶部(1)
430
平文分割部
CRT合成部
出力部
428
429
420 RSA署名装置
署名文

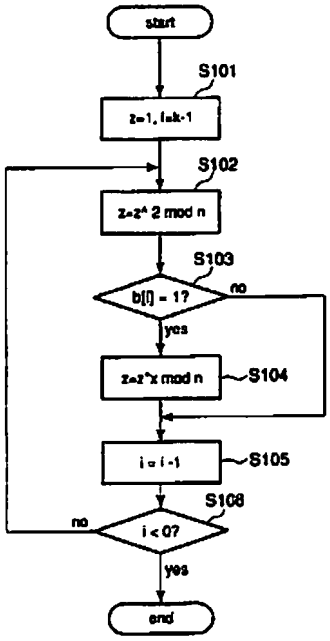
【図１０】


分割鍵生成装置から
331
分割鍵記憶部
332
べき剰余計算部
338
データ記憶部(2)
334
データ記憶部(1)
333
データ記憶部(0)
CRT合成部
336
出力部
337
署名文
330 RSA署名装置
平文
平文分割部
335

【図１１】


start
Sp=M mod p, Sq=M mod q
i=0, j=0, u=0, t=0
S31'
S32'
tt[i]?
01
10 or 11
00
S33
S35
S40
Sp = Sp*dp[j] mod p
Sq = Sq*dq[u] mod q
S' = Sp*du[t] mod p (tt[i]=10 のとき)
or
S' = Sq*du[t] mod q (tt[i]=11 のとき)
S34
S36
j = j + 1
u = u + 1
t = t + 1
S37
S41
i = i + 1
S38'
no
i > (kp+kq+x)?
yes
S39
S = Sp*q*(q^-1 mod p)+
Sq*p*(p^-1 mod q)(mod n)
end

【図１２】


start
S101
z=1, i=k-1
S102
z=z^ 2 mod n
S103
b[i] = 1?
no
yes
z=z*x mod n
S104
i = i -1
S105
S106
i < 0?
no
yes
end

フロントページの続き

(74)代理人 100070437
弁理士 河井 将次
(72)発明者 友枝 裕樹
神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内
(72)発明者 川村 信一
神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内
(72)発明者 新保 淳
神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内
Fターム(参考) 5J104 AA22 EA28 JA23 NA02 NA18